

CLAIMS

What is claimed is:

- Sub. B1
- 5
- 7
1. A method for clipping graphics primitives for display, comprising the steps of:
defining all vertices of a primitive using relational coordinates;
using a clipping algorithm with a circular buffer to store input and output polygons of said primitive; and
for each vertex of said polygons:
indicating whether it is visible with respect to each plane of a view volume.
 2. The method of Claim 1, wherein only vertices which are visible in all said planes are rasterized.
 3. The method of Claim 1, wherein said using step is performed prior to any lighting, fog, or texture calculations.
 4. The method of Claim 1, wherein said primitive is a polygon.
 5. The method of Claim 1, wherein said primitive is a triangle.
 6. The method of Claim 1, wherein said relational coordinates are barycentric.
 7. The method of Claim 1, wherein said view volume is a frustum.
- Sub. B2

8. The method of Claim 1, wherein there are six or more planes in said view volume.
9. The method of Claim 1, wherein there are six planes in said view volume.
10. The method of Claim 1, wherein there are twelve planes in said view volume.
11. The method of Claim 1, wherein said clipping algorithm is the Sutherland and Hodgman polygon clipping algorithm.
12. The method of Claim 1, wherein vertex visibility in each of said planes is indicated by a bit flag.
13. The method of Claim 1, wherein vertex visibility is indicated by twelve bit code.
14. The method of Claim 1, wherein two circular buffers are used to store said input and output polygons.
15. The method of Claim 1, wherein said circular buffer has a maximum storage of sixteen vertices.

- 59bA
- 5
16. A geometry unit, comprising:
circuitry to define all vertices of a primitive using relational coordinates;
circuitry to implement a clipping algorithm with a circular buffer to store input and output polygons of said primitive; and
circuitry to set an outcode value for each said vertex indicating whether it is visible with respect to each of plane of said view volume.
17. The geometry unit of Claim 16, wherein only vertices which are visible in all said planes are rasterized.
18. The geometry unit of Claim 16, wherein said clipping algorithm is implemented prior to any lighting, fog, or texture calculations.
19. The geometry unit of Claim 16, wherein said primitive is a polygon.
20. The geometry unit of Claim 16, wherein said primitive is a triangle.
21. The geometry unit of Claim 16, wherein said relational coordinates are barycentric.
22. The geometry unit of Claim 16, wherein said view volume is a frustrum.
23. The geometry unit of Claim 16, wherein there are six or more planes in said view volume.

24. The geometry unit of Claim 16, wherein said clipping algorithm is the Sutherland and Hodgman polygon clipping algorithm.

25. The geometry unit of Claim 16, wherein vertex visibility in each of said planes is indicated by a bit flag.

26. The geometry unit of Claim 16, wherein two circular buffers are used to store said input and output polygons.

27. The geometry unit of Claim 16, wherein said circular buffer has a maximum storage of sixteen vertices.

28. A pipelined graphics system, comprising:

a transformation unit connected to:

transform a primitive into a clipping space, and

assign a fixed barycentric coordinate to each vertex of said primitive; and

a geometry unit connected to:

perform clip testing on said primitives,

clip said primitives, if necessary, according to said fixed barycentric coordinates, and

set an outcode value for each said vertex indicating whether it is visible with respect to each plane of a view volume.

29. The pipelined graphics system of Claim 28, wherein only vertices which are visible in all said planes are rasterized.

30. The pipelined graphics system of Claim 28, wherein said clipping is implemented prior to any lighting, fog, or texture calculations.

31. The pipelined graphics system of Claim 28, wherein said primitive is a triangle.
32. The pipelined graphics system of Claim 28, wherein said view volume is a frustrum.
33. The pipelined graphics system of Claim 28, wherein there are six or more planes in said view volume.
34. The pipelined graphics system of Claim 28, wherein said clipping uses the Sutherland and Hodgman polygon clipping algorithm.
35. The pipelined graphics system of Claim 28, wherein vertex visibility in each of said planes is indicated by a bit flag.
36. The pipelined graphics system of Claim 28, wherein two circular buffers are used to store said input and output polygons.
37. The pipelined graphics system of Claim 28, wherein said circular buffer has a maximum storage of sixteen vertices.

38. A computer system comprising:

display hardware;

a processor connected to provide graphics data

a geometry and lighting accelerator connected to receive said

5 graphics data, comprising

a transformation unit connected to:

transform a primitive into a clipping space, and

assign a fixed barycentric coordinate to each vertex of said

primitive; and

10 a geometry unit connected to

perform clip testing on said primitives,

clip said primitives, if necessary, according to said fixed

barycentric coordinates,

15 set an outcode value for each said vertex indicating whether it

is visible with respect to each plane of a view volume, and

output clipped graphics data to be rendered; and

video rendering hardware connected to receive said clipped graphics

data, generate graphics, and connected to display said graphics

on said display hardware.

39. The computer system of Claim 38, wherein said clipping is implemented prior to any lighting, fog, or texture calculations.

40. The computer system of Claim 38, wherein said primitive is a triangle.

41. The computer system of Claim 38, wherein said clipping uses the Sutherland and Hodgman polygon clipping algorithm.

42. The computer system of Claim 38, wherein vertex visibility in each of said planes is indicated by a bit flag.
43. The computer system of Claim 38, wherein two circular buffers are used to store said input and output polygons.
44. The method of Claim 38, wherein said video rendering hardware is a GLINT 500TX.
45. The method of Claim 38, wherein said video rendering hardware is a GLINT PERMEDIA.
46. The method of Claim 38, wherein said processor is an X86-compatible processor.

App'n of 3Dlabs Inc., Ltd.: TD-143 Page 75